

# Introduction of sigstore

Masayuki Imanishi

Norio Kobota

<https://sigstore.dev/>

Linux Foundation傘下のPJとして、ソフトウェアの来歴と完全性を証明するサービスの提供を開始(2021/3/9発表)

[https://sigstore.dev/what\\_is\\_sigstore/](https://sigstore.dev/what_is_sigstore/)

## □ 現在のソフトウェアサプライチェーン上の課題

- ✓ ソフトウェアの電子署名を行うために必要な鍵の管理、証明書の更新、失効、公開鍵の配布を安定して運用することが難しい。
- ✓ ソフトウェアのダイジェストをサーバに公開している場合も多いが、ダイジェストのすり替えやユーザへの標的型攻撃などさまざまな攻撃を受ける可能性がある。

## □ sigstoreの解決方法

- ✓ sigstoreが**透明性ログ管理サービス**(rekor)、PKI(fulcio, Private CA)を提供
- ✓ **証明書の有効期限を20分**にすることで、鍵の管理、失効の手間を軽減
- ✓ OpenID Connect接続成功時のメールアドレスを証明書に含むことで、署名時にメールアドレスが有効であったことを証明可能  
⇒ OpenID Connectを使うことで、2FA, OTP等と組み合わせることも可能
- ✓ 電子署名は透明性のあるログ管理技術を使って管理



<https://www.linuxfoundation.org/press-release/2021/03/linux-foundation-announces-free-sigstore-signing-service-to-confirm-origin-and-authenticity-of-software/>

### What is sigstore?

**sigstore** will be a free to use non-profit software signing service that harnesses existing technologies of x509 PKI and transparency logs.

Users generate ephemeral short-lived key pairs using the sigstore client tooling. The sigstore PKI service will then provide a signing certificate generated upon a successful OpenID connect grant. All certificates are recorded into a certificate transparency log and software signing materials are sent to a signature transparency log. The use of transparency logs introduces a trust root to the users OpenID account. We can then have guarantees that the claimed user was in control of an identity service providers account at the time of signing. Once the signing operation is complete, the keys can be discarded, removing any need for further key management or need to revoke or rotate.

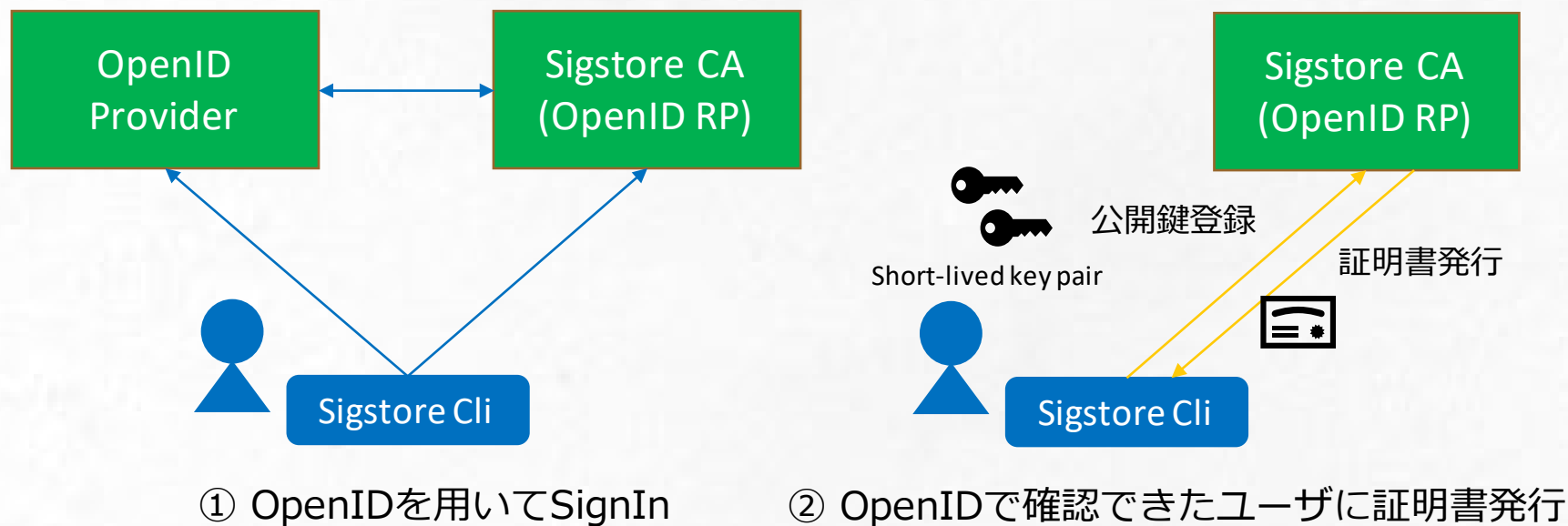
Using OpenID connect identities allows users to take advantage of existing security controls such as 2FA, OTP and hardware token generators.

sigstore's transparency logs can act as a source of provenance, integrity, and discoverability. Being public and open anyone can monitor sigstore's transparency logs for occurrences of their software namespace being used, perform queries using an artifact's digest, return entries signed by a specific email address, public key, etc. Further to this, security researchers can monitor the log to seek out possible nefarious patterns or questionable behavior.

[https://sigstore.dev/what\\_is\\_sigstore/](https://sigstore.dev/what_is_sigstore/)

# sigstore signing flow

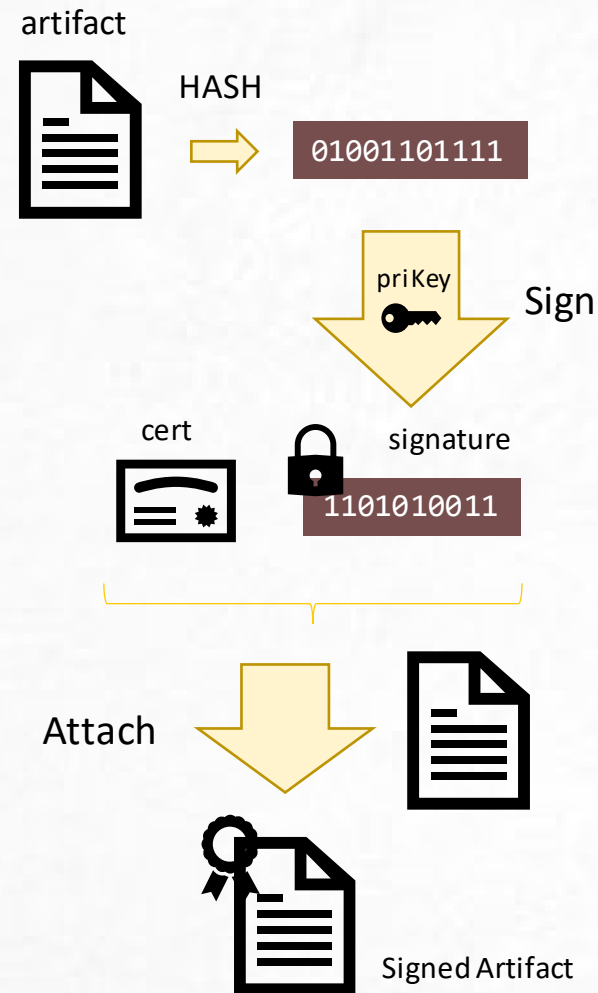
※OpenID Connect(OIDC)を使った場合のケース



OTPとか、2FAとか既存のセキュリティ機構が使えるのメリットだね

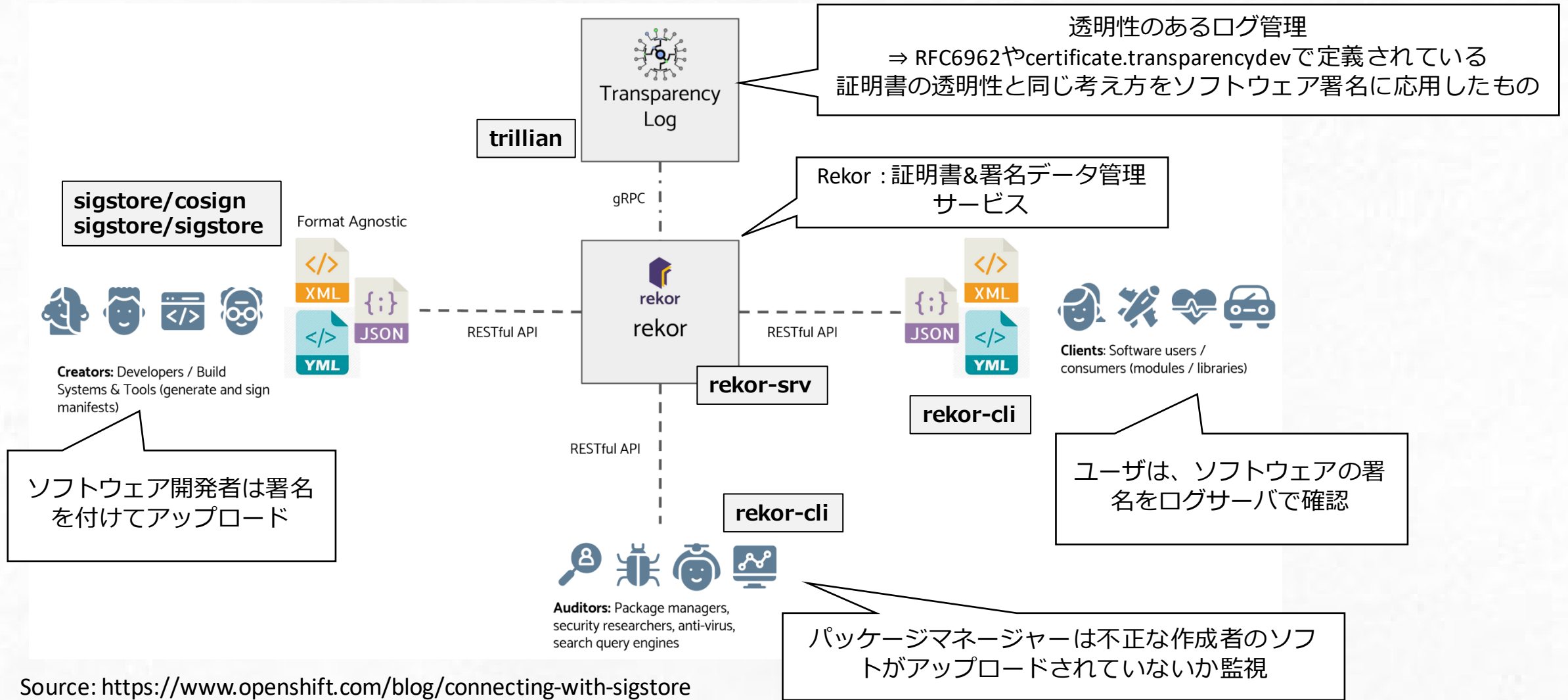
おまけ

④ 発行した証明書全と、sigstore CliのSign記録は、すべてsigstoreが管理するログサーバへと送られ保管されるので、key pairが無くなろうと、証明書のrevokeとかは必要ない。



③ ②の秘密鍵と証明書を使って署名

# sigstore Architecture



# Google Trillian

sigstoreのバックエンドには、googleが開発したTrillianを使用

<https://github.com/google/trillian>

Trillianはマークルツリーと電子署名を使って、ログの改ざんを困難にするソフトウェア

ユースケースの一つが証明書の透明性確保

<https://certificate.transparency.dev/>

IETF RFC6962 Certificate Transparency

<https://tools.ietf.org/html/rfc6962>

sigstoreのユースケースは**証明書の透明性**だけでなく、ソフトウェアの**署名の透明性確保**のために利用

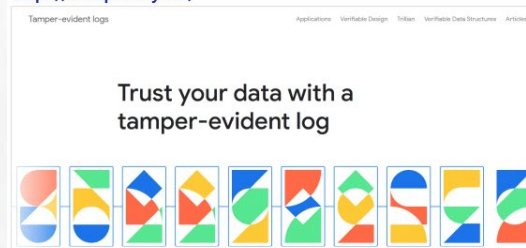
Googleが開発した「改ざん不可能なログシステム」を構築できる「Trillian」とは？



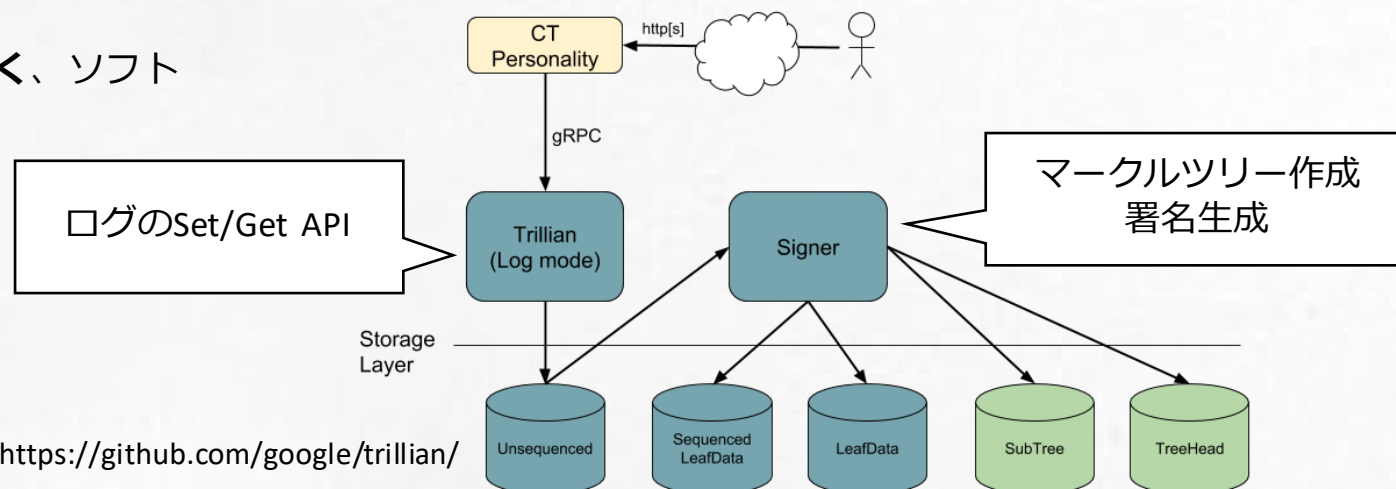
システムログは悪意のある第三者の攻撃や不正を監査するのに役立ちますが、ログそのものを改ざんされてしまつては意味がありません。Googleが開発する「Trillian」は、そんなログの改ざんをハッシュ木を用いて困難にした、オープンソースのログ基盤です。

An open-source append only ledger | Trillian

<https://transparency.dev/>



<https://gigazine.net/news/20210219-google-certificate-transparency-trillian/>



Source : <https://github.com/google/trillian/>

# 証明書の透明性 (RFC6962)

<https://certificate.transparency.dev/>

IETF RFC6962 Certificate Transparency

<https://tools.ietf.org/html/rfc6962>

## ■ 課題

認証局による誤発行や攻撃者による不正な発行

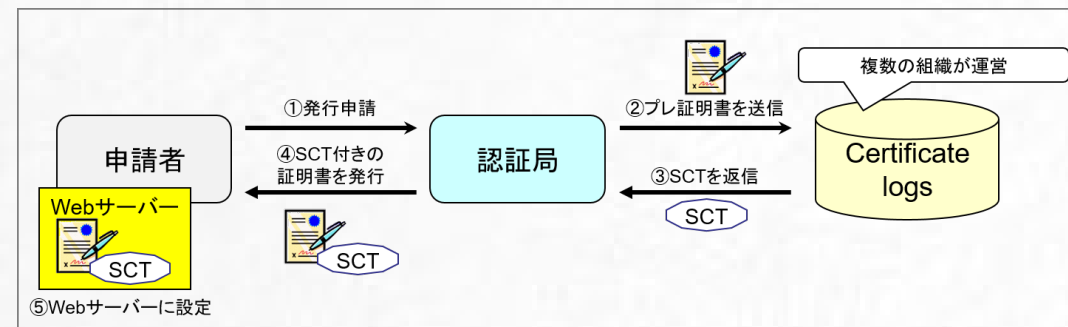
例えば2015年にSymantecの子会社のThawte社が  
内部テストで[www.google.com](http://www.google.com)のEV SSL証明書を勝手に発行

⇒このときはログサーバに上記の証明書が登録されていることで発覚

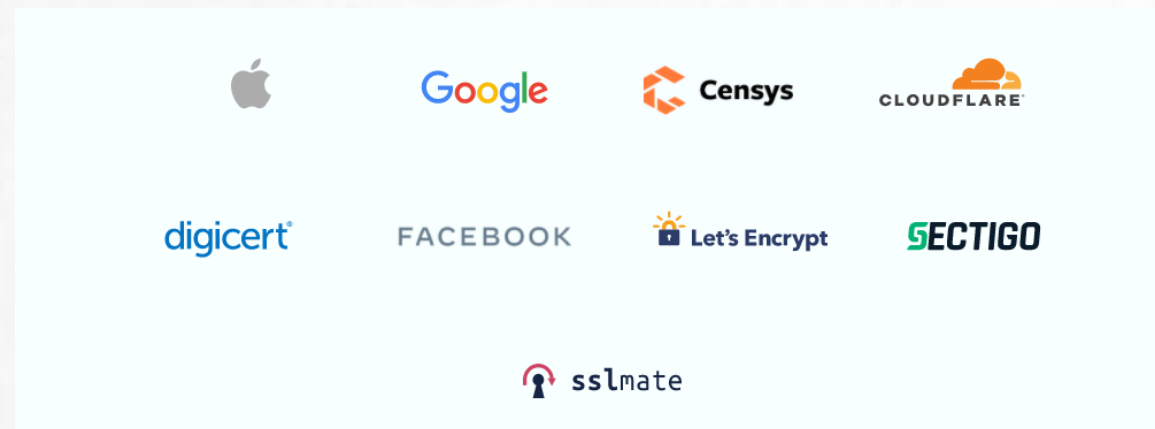
## ■ 解決策

認証局が証明書発行時にログサーバに送付し、  
ログサーバでSCT署名を付与し、ログサーバに残す。  
ブラウザ側はサーバアクセス時にSCTのオプションを見て、  
それがログサーバに登録されているものか検証可能

ログサーバのデータはマークルツリーを使って署名を行う

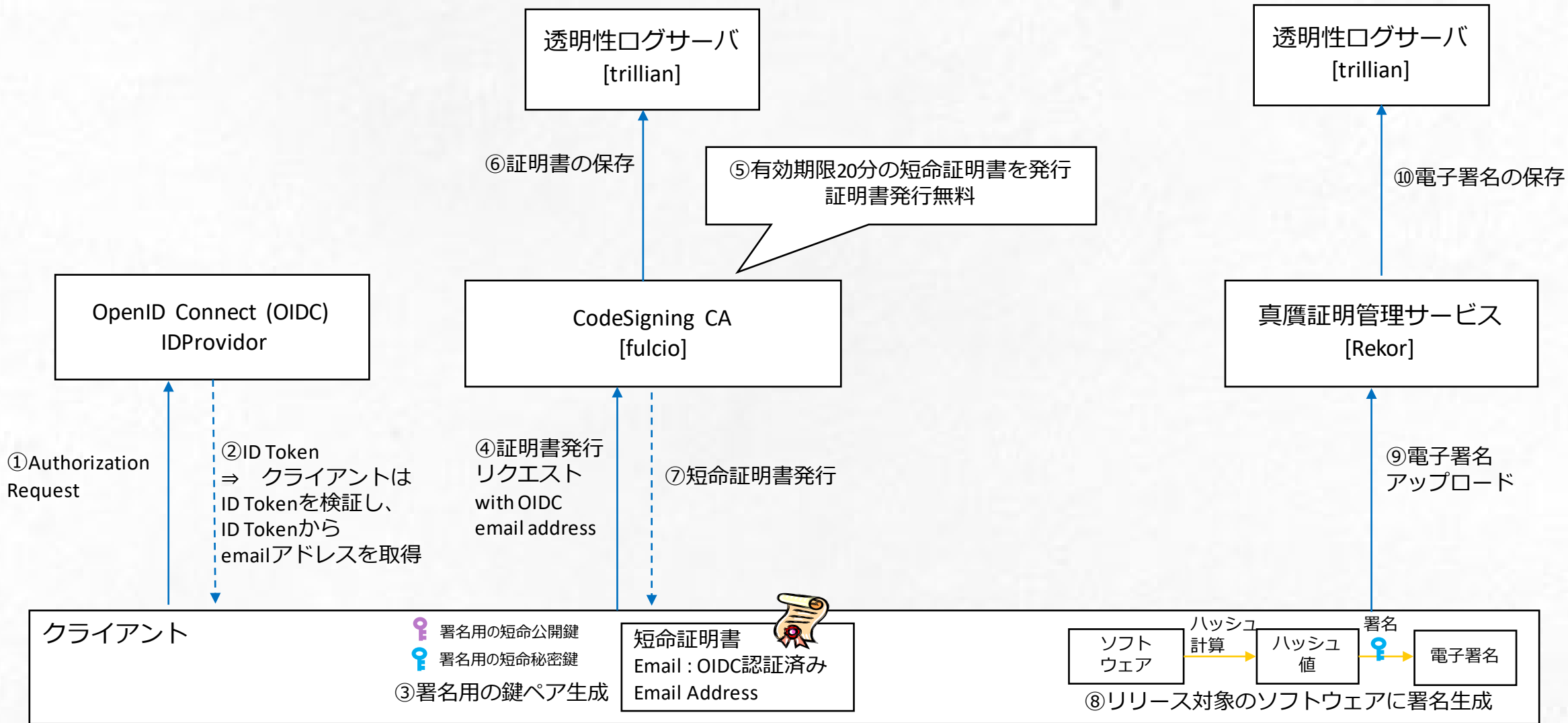


Source : <https://jprs.jp/glossary/index.php?ID=0235>



# sigstore System

[]はOSS名称





# sigstoreと、現在の署名の比較

	sigstore OpenID Connectを使うケース	[現在のソフトウェア署名のやり方] Public CAから証明書を購入するケース
認証局	Private CA	Public CA
証明書の価格	○ 無料	× 高い(約\$200/ year)
証明書の有効期限	20分	1-3年
認証局に対する信頼性。第三者監査	× 認証局に対して第三者監査を行っているかどうかの記載は見つからなかった	○
秘密鍵管理の手間	○署名を行うたびに秘密鍵を生成し、署名と登録が終われば秘密鍵を破棄すればよい	× 証明書の有効期限の間、秘密鍵を安全に保管する必要がある
鍵が漏えいしたときの対応	○短期間の証明書なのでRevokeの必要なし	×認証局に対してRevokeの処理が必要
タイムスタンプ局	オプションでコンテンツがその時間に存在したことを証明するために利用可能 アマノだと1スタンプ8円	
セキュリティモデル	OIDCトークンが電子メールアドレスの所有者の証明。 すべての署名と証明書をログに登録、公開する。 クライアントは、ログにない署名と証明書を信頼しない	信頼されたPublic CAから発行された証明書を使って署名を行う。 クライアントは、OSに入っている信頼された認証局証明書を使って検証する



# まとめ

- sigstoreは、Linux Foundation傘下のPJで、ソフトウェアの来歴と完全性を証明するサービスを提供
- 「有効期限20分の短命証明書」「透明性のあるログ管理サービス」「OpenID Connect」を組み合わせることで、ソフトウェア開発者の秘密鍵管理のコストを低減しながら、ソフトウェアの来歴証明が可能。
- sigstore運用に関する第三者監査を行うかの記載は見つからないので、正しく運用されていくのかは引き続き動向を見ていきたい。
- 現状はcontainerに対する署名だけ利用可能だが、SPDXドキュメントなど、鍵管理の難しいアーティファクトに対する汎用署名としての応用可能性を考えたい。

# Any Questions?